IN THE CLAIMS

Following are the claims as amended herein and as currently pending for consideration:

1. (Currently Amended) A method for performing a bi-linear interpolation or a motion compensation of a digital image or video, the method comprising:

decoding a first shuffle instruction and a first multiply-add instruction, each of an instruction format comprising a first operand field and a second operand field;

responsive at least in part to said first shuffle instruction, generating a first packed data having a first plurality of byte data elements including an a_1 byte data element, and at least two copies of each of a_2, a_3, and a_4 byte data elements; and

responsive to said first multiply-add instruction, wherein the first operand field of said first multiply-add instruction specifies said first packed data and the second operand field specifies a second packed data having a second plurality of byte data elements including at least two copies of each of b_1 and b_2 byte data elements, performing an operation $(a_1 \times b_1) + (a_2 \times b_2)$ to generate a first 16-bit data element of a third packed data, performing an operation $(a_2 \times b_1) + (a_3 \times b_2)$ to generate a second 16-bit data element of the third packed data, and performing an operation $(a_3 \times b_1) + (a_4 \times b_2)$ to generate a third 16-bit data element of the third packed data.

2. (Original) The method of Claim 1, further comprising:

decoding a second shuffle instruction and a second multiply-add instruction, each of an instruction format comprising a first operand field and a second operand field;

responsive at least in part to said second shuffle instruction, generating a fourth

packed data having a fourth plurality of byte data elements including a c1 byte data element, and at least two copies of each of c2, c3, and c4 byte data elements; and

responsive said second multiply-add instruction, wherein the first operand field of said second multiply-add instruction specifies said fourth packed data and the second operand field of said second multiply-add instruction specifies a fifth packed data having a fifth plurality of byte data elements including at least two copies of each of d1 and d2 byte data elements, performing an operation (c1 x d1) + (c2 x d2) to generate a first 16-bit data element of a sixth packed data, performing an operation (c2 x d1) + (c3 x d2) to generate a second 16-bit data element of the sixth packed data, and performing an operation (c3 x d1) + (c4 x d2) to generate a third 16-bit data element of the sixth packed data.

3. (Original) The method of Claim 2, further comprising:

decoding a packed add instruction of an instruction format comprising a first operand field and a second operand field;

responsive to said packed add instruction, wherein the first operand field of said packed add instruction specifies said third packed data and the second operand field of said packed add instruction specifies the sixth packed data, adding the first 16-bit data elements of the third and sixth packed data to generate a first 16-bit data element of a seventh packed data, adding the second 16-bit data elements of the third and sixth packed data to generate a second 16-bit data element of the seventh packed data, and adding the third 16-bit data elements of the third and sixth packed data to generate a third 16-bit data element of the seventh packed data.

4. (Original) The method of claim 1, said first plurality of byte data elements including at least 16 byte data elements and said second plurality of data elements including at least 16 byte data elements.

5. (Previously Presented) The method of claim 1, said first plurality of byte data elements including 8 byte data elements and said second plurality of data elements including 8 byte data elements

6. (Original) The method of claim 1 wherein said first plurality of data elements are treated as unsigned bytes.

7. (Original) The method of claim 6 wherein said second plurality of data elements are treated as signed bytes.

8. (Original) The method of claim 7 wherein each of said first, second and third 16-bit data elements are generated using signed saturation.

9. (Original) An apparatus to perform the method of Claim 8 comprising:

an execution unit including one or more execution circuits to execute operations on packed data elements;

at least one state machine; and

a machine-accessible medium including data that, when accessed by said at least one state machine, causes said at least one state machine to enable the one or more execution circuits to perform the method of Claim 9.

10. (Original) The method of claim 1 wherein said first operand field comprises bits five through three of the instruction format.

11. (Original) The method of claim 10 wherein said second operand field comprises bits two through zero of the instruction format.

12. (Original) The method of claim 11 wherein said first plurality of byte data elements is overwritten by said third packed data responsive to the first multiply-add instruction.

13. (Currently Amended) A machine-accessible medium including data to perform a bi-linear interpolation or a motion compensation of a digital image or video such that, when accessed by one or more machines, causes said one or more machines to:

shuffle a first 2n+1 byte elements of a first line of data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements;

shuffle a second 2n+1 byte elements of a second line of data to generate a second packed data comprising at least a second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements;

multiply-add the first packed data with at least a first two byte coefficients to generated a third packed data including sums of products;

multiply-add the second packed data with at least a second two byte coefficients to generated a fourth packed data including sums of products; and

add corresponding sums of products of the third and fourth packed data to

generate a first packed result.

14. (Original) The machine-accessible medium of claim 13 including data that, when accessed by said one or more machines, causes said one or more machines to treat elements of the first packed data and of the second packed data as unsigned bytes in generating the sums of products of the third packed data and of the fourth packed data respectively.

15. (Original) The machine-accessible medium of claim 14 including data that, when accessed by said one or more machines, causes said one or more machines to treat elements of said at least the first two byte coefficients and of said at least the second two byte coefficients as signed bytes in generating the sums of products of the third packed data and of the fourth packed data respectively.

16. (Original) The machine-accessible medium of claim 14 including data that, when accessed by said one or more machines, causes said one or more machines to overwrite the first packed data with the third packed data and to overwrite the second packed data with the fourth packed data.

17. (Original) The machine-accessible medium of Claim 13 including data that, when accessed by said one or more machines, further causes said one or more machines to:

shuffle the second $2n+1$ byte elements of the second line of data to generate the first packed data comprising at least the second $4n$ byte elements including $2n-1$ duplicated elements of the second $2n+1$ byte elements; and

shuffle a third 2n+1 byte elements of a third line of data to generate the second packed data comprising at least a third 4n byte elements including 2n-1 duplicated elements of the third 2n+1 byte elements.

18. (Original) The machine-accessible medium of Claim 17 wherein n is equal to 2 and including data that, when accessed by said one or more machines, further causes said one or more machines to:

generate a second packed result comprising at least 4n rounded averages, each corresponding to an element of said first packed result.

19. (Original) An apparatus comprising:

a decoder to decode a plurality of instructions including a shuffle instruction and a multiply-add instruction;

an execution unit including a first execution circuit, enabled by the decoded shuffle instruction, to shuffle a first 2n+1 byte elements of a first line of data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements, said execution unit further including a second execution circuit, enabled by the decoded multiply-add instruction, to multiply each of a first pair of byte data elements of the first packed data with a first pair of respective byte coefficient elements and to generate a first 16-bit result representing a first sum of products of the first pair of multiplications, and to multiply each of a second pair of byte data elements of the first packed data with a second pair of respective byte coefficient elements and to generate a second 16-bit result representing a second sum of products of the second pair of multiplications;

a first register to store the first packed data in response to the shuffle instruction; and

a second register to store a third packed data comprising at least said first and second 16-bit results in response to the multiply-add instruction.

20. (Original) The apparatus of claim 19 wherein n is at least two.

21. (Original) The apparatus of claim 20 wherein said first packed data contains at least sixteen byte data elements.

22. (Original) The apparatus of claim 19 wherein the first packed data comprises unsigned byte data elements.

23. (Original) The apparatus of claim 22 wherein said first and second pairs of respective byte coefficient elements comprise signed byte data elements.

24. (Original) The apparatus of claim 19 wherein the first and second 16-bit results are generated using signed saturation.

25. (Original) A computing system comprising:

an addressable memory to store data;

a processor including:

a first storage area to store byte data elements of a content data;

a second storage area to store byte coefficient elements;

a decoder to decode a plurality of instructions including a first and a second shuffle instruction and a first and a second multiply-add instruction;

a first execution circuit, enabled by the decoded first shuffle instruction, to shuffle a first 2n+1 byte elements of the content data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements, and enabled by the decoded second shuffle instruction, to shuffle a second 2n+1 byte elements of the content data to generate a second packed data comprising at least a second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements;

a second execution circuit, enabled by the decoded first multiply-add instruction, to multiply each of a first and second pair of byte data elements of the first packed data with respective byte coefficient elements of a third packed data and to generate a first and a second 16-bit result respectively representing a first and a second sum of products of the first and the second pair of multiplications, and enabled by the decoded second multiply-add instruction, to multiply each of a third and fourth pair of byte data elements of the second packed data with respective byte coefficient elements of a fourth packed data and to generate a third and a fourth 16-bit result respectively representing a third and a fourth sum of products of the third and the fourth pair of multiplications;

a third storage area to store packed 16-bit data elements including the first and the second 16-bit results responsive to the first multiply-add instruction; and

a fourth storage area to store packed 16-bit data elements including the third and the fourth 16-bit result responsive to the second multiply-add instruction; and

a storage device to store said plurality of instructions.

26. (Original) The computing system of claim 25 wherein each of the first, second, third and fourth packed data comprise 16 byte data elements.

27. (Original) The computing system of claim 25 wherein each of the first, second, third and fourth packed data comprise 8 byte data elements.

28. (Original) The computing system of claim 25 wherein each of said first, second, third and fourth 16-bit results are generated by multiplying unsigned bytes elements of the content data by signed coefficient elements and by adding pairs of products with signed saturation.

29. (Original) The computing system of claim 25 wherein said plurality of instructions further includes an unpack instruction to store the byte data elements of the content data in the first storage area and a packed add instruction to add corresponding 16-bit data elements of the third and fourth storage areas.